(2)

AD-A251 862

DTIC
ELECTE
JUN 2 1992
S C D

QUARTERLY REPORT
1 Oct 91 – 31 Mar 92

Using Computer Design and Simulation to
Improve Manufacturing Productivity

ARPA Order Number:	6419
Program Code Number:	8E20
Contractor:	Cornell University
Contract Number:	ONR N00014-88-K-0591
Contract Amount:	$2,931,739
Effective Date of Contract:	July 1, 1988
Expiration Date of Contract:	September 30, 1991
Principal Investigator:	John E. Hopcroft
Telephone Number:	607/255-7316
Email Address:	jeh@cs.cornell.edu
Reporting Period:	1 Oct 91 – 31 Mar 92

92-14180

92 5 28 141

1 Description of Progress

The members of the Modeling and Simulation Group at Cornell believe that improvements in computer capabilities and in software design tools have the potential to radically alter the manufacturing process. These improvements present unique opportunities for improving manufacturing productivity, an area of strategic importance to our nation's defense and economic future. Potential payoffs to industry and to the military include extended capabilities, increased reliability of components, and reduced production costs.

Our goal is to develop and extend the science base in representing, manipulating, and reasoning about physical objects and processes. To this end, we are carrying out a broad-based research program to develop the underlying methodology that supports representation, editing, and simulation of objects, tasks, and systems.

The remainder of this report summarizes some of our work in one of the areas that we stressed in our original proposal: robust and reliable geometric algorithms.

1.1 Robust and Reliable Geometric Algorithms

We are investigating how to design geometric algorithms, like those used in solid modelers and finite-element mesh generators, that are robust. An implementation of an algorithm is *robust* if it fails only when the underlying hardware fails. Most geometric algorithms lack robustness since they are designed assuming infinite precision arithmetic. Roundoff error or imprecise input can cause these algorithms to fail.

Robust Polyhedral Intersection

James Stewart, a recent Ph.D. graduate supported by our project, has been developing algorithms for robust polyhedral intersection. A basic question that must be resolved when intersecting two polyhedra is where an edge of one polyhedron intersects the plane of a face of the other polyhedron; if the intersection point is *within* the face then it is common to both polyhedra and must be a vertex of the polyhedron of intersection.

In our previous reports the notion of an *approximate polygon* has been used to model polygons whose positions were uncertain. Using approximate polygons, Stewart has developed a robust algorithm which answers the point location question. This algorithm is fast and simple, and can handle uncertainty in both the location of the point and the polygon location. The algorithm can be used to build robust intersection algorithms, and hence, more robust solid modelers.

In related work, Dr. Stewart has concluded that formal correctness is nearly impossible to achieve in a robust polyhedral intersection algorithm, and that proving *local correctness* is the best alternative. A locally correct algorithm always succeeds when presented with localized special cases; for example, a locally correct intersection algorithm always succeeds when intersecting two corners, two edges, or two faces.

Dr. Stewart has derived a set of *robustness rules* which, when incorporated in an intersection algorithm, cause the algorithm to be locally correct for intersections of two corners and intersections of two edges. Preliminary versions of this algorithm have been tested and found to be significantly more robust than earlier designs.

Implicit Patches for Surface Representaion

A useful geometric modeling system must provide a user with the ability to design and to manipulate three dimensional models represented by free-form surfaces. Such models are important when modeling mechanical parts, medical implants, and plastic flow in molds. Traditionally, free-form surfaces are built from parametric patches. Parametric patches have been successful for both design and rendering, but the manipulation of three dimensional models composed of parametric patches poses fundamental difficulties. For example, parametric patches are not closed under sweeping or convolution, and the intersection of a pair of parametric patches is extremely difficult to represent and evaluate.

One way to overcome the above difficulties is to build free-form surfaces from low-degree implicit patches. Implicit patches are closed under all common operations in geometric modeling, and the intersections of low-degree implicit patches can be computed efficiently. Implicit surfaces are also very useful in deriving blending surfaces. Recent research indicates that quadric and cubic implicit patches are versatile enough to build arbitrary three dimensional models.

A major reason for the popularity of parametric patches has been their good shape cont ' properties. In recent work, Baining Guo, a recent Ph.D. graduate supported by our project, has tackled the shape control issues of implicit patches. Using Bernstein-Bezier representation of polynomials, the shapes of implicit patches can be controlled by manipulating their control points. In designing free-form surfaces, one often encounters various shape requirements, such as a nice pattern of reflection lines or restrictions on the minimum radius of curvature. Among all the shape requirements, convexity is the most basic and the most frequently requested. Guo has shown how to manipulate the control points of a quadric patch or a cubic patch so that it becomes convex.

Guaranteed-Quality Mesh Generation

Mesh generators are useful for the numerical solution of partial differential equations (PDEs). Such numerical solutions are needed for a wide variety of engineering problems including, for example, fluid dynamics, structural analysis, thermal analysis, and electromagnetics. For several commonly-used PDE solution techniques, the first step is to divide the problem region into simply-shaped elements creating a *mesh*.

Our research has led to the development and implementation of a mesh generator for 2-dimensional (2D) problems and for curved surfaces that is guaranteed (by mathematical proof) to exhibit the following properties:

- Internal and external boundaries are respected.
- Element shapes are guaranteed. All elements are triangles with angles between 30 and 120 degrees (except for smaller angles that are required as part of the boundary).
- Element density can be controlled. Small elements lead to more accurate solutions, but require more computing time. Thus, we want small elements in "interesting" regions and large elements elsewhere.

- In addition, our mesh generator is fast. Many existing mesh generators use an iterative smoothing process to improve an initial mesh. For our mesh generator, shapes are guaranteed without smoothing.

Our goal is to create guaranteed-quality mesh generators for 2D regions, for regions on curved surfaces, and for 3D regions. The 2D version has been implemented. A mesh generator for a simple curved surface (a single parametric patch) has been implemented; we are currently extending this version to handle surfaces defined by multiple parametric patches. For full 3D problems, we have some partial results, but our current techniques cannot guarantee the shape of the resulting tetrahedra.

2 Presentations, Publications, and Reports

Presentations

- *Equate: An Object-Oriented Constraint Solver*, ACM Conference on Object-Oriented Programming Systems, Languages and Applications, Phoenix, October 1991. Michael Wilk.
- *Guaranteed-Quality Meshes for Curved Surfaces*, Workshop on Computational Geometry, Mathematical Sciences Institute, SUNY Stony Brook, October 1991. Paul Chew.
- *Issues in Physical System Simulation — Integrating Physics Model Specification, Geometry, Symbolic Mathematics, and Control*, Minisymposium 16 of the Second SIAM Conference on Geometric Design, Tempe, Arizona, November 1991. Session organized by Jim Cremer.
- *SimLab: An Environment for High Level Specification and Automatic Generation of Physical Systems Simulators*, Second SIAM Conference on Geometric Design, Tempe, Arizona, November 1991. Richard S. Palmer.
- *Specifying and Generating Simulators with SimLab: Some Examples*, Second SIAM Conference on Geometric Design, Tempe, Arizona, November 1991. Jim Cremer.
- *Symbolic/Numeric Techniques in Modeling and Simulation*, University of Illinois, Urbana-Champagne, November 1991. Richard Zippel.
- *Symbolic Computation using Weyl*, Supercomputing Research Center, Maryland, December 1991. Richard Zippel.
- *Creating Simulators Directly From Models of Physics*, DARPA Manufacturing Technology Contractors Meeting, Salt Lake City, January 1992. Jim Cremer.
- *Symbolic Computation using Weyl*, Department of Computer Science, Tel Aviv University, Isreal, March 1992. Richard Zippel.
- *Symbolic/Numeric Techniques in Modeling and Simulation*, Dept. of Electrical Engineering, Technion, Haifa, Israel, March 1992. Richard Zippel.
- *Symbolic Computation using Weyl*, IBM Scientific Center, Haifa, Israel, March 1992. Richard Zippel.
- *Applications of Geometric Algorithms*, AFOSR Workshop, Purdue Univerisity, West Lafayette, Indiana, March 1992. Paul Chew.
- *Software for Design in Manufacturing*, DARPA Software Technology Conference, Los Angeles, April 1992. Richard Zippel.

Publications

1. *Computer Algebra and Parallelism II, May 1990 MSI Workshop Proceedings*, Springer-Verlag, Lecture Notes in Computer Science v.584, 1992. Edited by Richard Zippel.

2. Equate: An Object-Oriented Constraint Solver, *Proceedings of the ACM Conference on Object-Oriented Programming Systems, Languages and Applications*, ACM Press, 1991, 286–298. Michael R. Wilk.

3. Masking Failures of Multidimensional Sensors, *Proceedings of the Tenth Symposium on Reliable Distributed System*, October 1991, 32–41. Paul Chew and Keith Marzullo.

4. Programming Mechanical Simulations, *Proceedings of the Second Eurographics Workshop on Animation and Simulation*, September 1991, 223–243. J. K. Kearney, S. Hansen, and J. F. Cremer.

Technical Reports

5. SIMLAB: *Automatically Creating Physical Systems Simulators*, Department of Computer Science Tech Report 91–1246, November 1991. Richard S. Palmer and James F. Cremer.

6. *The Data Structure Accelerator Architecture*, Department of Computer Science Tech Report 91–1256, Cornell University, December 1991. Richard Zippel.

7. *On Dexterous Rotations of Polygons*, Department of Computer Science Tech Report 91–1258, Cornell University, December 1991. Daniela Rus.